CLAIMS

1. **(Currently Amended)** A method[[,]] comprising:

receiving an input of data, the input data conforming ~~that conforms~~ to a query language used by a filter engine comprising two or more filter sub-engines, wherein at least one filter sub-engine is a general filter sub-engine and at least one filter sub-engine is an optimized filter sub-engine;

determining whether the input data conforms to a grammar associated with the optimized filter sub-engine, wherein the optimized filter sub-engine is configured to handle only a subset of the query language handled by the general filter sub-engine;

in an event the determining indicates the input data conforms to the grammar associated with the optimized filter sub-engine:

determining whether the input data can be processed by the optimized filter sub-engine, the determining comprising identifying if the input data comprises a subset of the query language; and

directing the input data to the optimized filter sub-engine for processing;

~~whether the input data can be processed by an optimized filter sub-engine, wherein the optimized filter sub-engine is configured to handle only a subset of the query language handled by the general filter sub-engine, wherein the subset of the language does not include all aspects of the language; and~~

~~if the determining indicates that the input can be processed by the optimized filter sub-engine, directing the input data to the optimized filter sub-engine for processing;~~

in an event the determining indicates that the input data cannot be processed by the optimized filter sub-engine:

determining whether the input data can be processed by a second optimized filter sub-engine, wherein the second optimized filter sub-engine is configured to handle only a subset of the query language, and wherein the subset of the query language that the second optimized filter sub-engine is configured to handle excludes the subset of the query language that the first optimized filter sub-engine is configured to handle; and

directing the input data to the second optimized filter sub-engine for processing;

~~if~~ in an event the determining indicates that the input cannot be processed by the second optimized filter sub-engine, ~~then~~ directing the input to the general filter sub-engine for processing, wherein the general filter sub-engine is configured to handle all aspects of the query language; and

processing the input to derive a result.

2. (Canceled)

3. (Canceled)

4. (Previously Presented) The method as recited in claim 1, wherein the query language comprises a query language based on eXtensible Markup Language (XML).

5. (Canceled)

6. (Currently Amended) The method as recited in claim 1, the method further comprising:

parsing the input to identify first and second sub-expressions;

determining whether the first sub-expression can be processed by the optimized filter sub-engine;

~~if~~ in an event the first sub-expression can be processed by the optimized filter sub-engine, ~~then~~ directing the first sub-expression to the optimized filter sub-engine for processing;

~~if~~ in an event the first sub-expression cannot be processed by the optimized filter sub-engine, directing the first sub-expression to the general filter sub-engine for processing;

determining whether the second sub-expression can be processed by the optimized filter sub-engine;

~~If~~ in an event the second sub-expression can be processed by the optimized filter sub-engine, directing the second sub-expression to the optimized filter sub-engine for processing; and

~~If~~ in an event the second sub-expression cannot be processed by the optimized filter sub-engine, directing the second sub-expression to the general filter sub-engine for processing.

7. **(Original)** The method as recited in claim 6, further comprising:

obtaining a result of the processing of the first sub-expression; and

processing the second sub-expression only if the result of the first sub-expression is true.

8. **(Currently Amended)** A filter engine[[,]] comprising:

an optimized filter sub-engine configured to accept an input that conforms to a language and process the input against a filter table associated with the optimized filter sub-engine, wherein the optimized filter sub-engine is configured to process only a subset of terms of the language, wherein the subset of terms of the language does not include all terms of the language;

a general filter sub-engine configured to accept the input and process the input against a filter table associated with the general filter sub-engine, wherein the general filter sub-engine is configured to process all terms of the input language; and

an analyzer configured to determine whether the input can be processed by the optimized filter sub-engine and, if so, direct the input to the optimized filter sub-engine for processing or, if not, direct the input to the general filter sub-engine for processing.

9. **(Previously Presented)** The filter engine as recited in claim 8, wherein the analyzer is further configured to analyze a new filter added to the filter engine and to determine an appropriate filter sub-engine with which to associate the new filter.

10. (Previously Presented) The filter engine as recited in claim 8, wherein the language is XPath.

11. (Previously Presented) The filter engine as recited in claim 8, wherein the analyzer is further configured to determine whether the optimized filter sub-engine can process the input by comparing the input to a grammar associated with the optimized filter sub-engine and determining whether the input consists of terms that are compatible with the grammar.

12. (Currently Amended) The filter engine as recited in claim 8, further comprising a sub-expression module that is configured to:

determine whether the input consists of different sub-expressions;

~~if~~in an event the input consists of different sub-expressions, directing each of the different sub-expressions contained in the input to the ~~analyzer ; and~~ analyzer, wherein the analyzer is further configured to determine whether each of the different sub-expressions can be processed by the optimized filter sub-engine and to direct each of the different sub-expressions to an appropriate filter sub-engine for processing.

13. (Previously Presented) The filter engine as recited in claim 12, wherein a first of the different sub-expressions is directed to the optimized filter sub-engine and a second of the different sub-expressions is directed to the general filter sub-engine.

14. (Previously Presented) The filter engine as recited in claim 8, wherein the optimized filter sub-engine comprises:

a first optimized filter sub-engine configured to process inputs that conform to a first subset of the language; and

a second optimized filter sub-engine configured to process inputs that conform to a second subset of the language;

wherein the first subset of the language is different from the second subset of the input language.

15. (Currently Amended) ~~One or more computer-readable storage media containing computer-executable instructions that, when executed, direct a computing system to:~~

A computer-readable storage medium encoded with instructions that, when executed by a processor of a device, cause the device to perform acts comprising:

determining ~~determine~~ an appropriate filter sub-engine to which an input message should be directed for processing against a set of queries;

processing the input message using an optimized filter sub-engine if the optimized filter sub-engine comprises a grammar that supports processing of the input message;

processing the input message in a general filter sub-engine if the optimized filter sub-engine grammar does not support processing of the input message; and

wherein:

the input message is in accordance with a query language;

the optimized filter sub-engine supports a subset, less than the whole, of the query language; and

the general filter sub-engine supports the entire query language.

16. (Currently Amended) The ~~one or more computer-readable storage media~~ computer-readable storage medium as recited in claim 15, further comprising computer-executable instructions that, when executed, direct the computing system to perform acts comprising:

accept input messages for both the optimized filter sub-engine and the general filter sub-engine by way of a single input means so that an input message sending application ~~does not have to~~ is not required to distinguish between the optimized filter sub-engine and the general filter sub-engine.

17. (Currently Amended) The ~~one or more computer-readable storage media~~ computer-readable storage medium as recited in claim 15, wherein the query language is XPath.

18. (Currently Amended) The ~~one or more computer-readable storage media~~ computer-readable storage medium as recited in claim 15, wherein the query language is an XML query language.

19. (Currently Amended) The computer-readable storage medium ~~one or more computer-readable storage media~~ as recited in claim 15, further comprising computer-executable instructions that, when executed, direct the computing system to perform acts comprising:

prior to determining which filter sub-engine will process the input message, parse the input message into two or more sub-expressions;

for each of the two or more sub-expressions, determine an appropriate filter sub-engine that can process the sub-expression; and

direct each of the two or more sub-expressions to the appropriate filter sub-engine for processing.

20. (Currently Amended) The computer-readable storage medium ~~one or more computer-readable storage media~~ as recited in claim 19, further comprising computer-executable instructions that, when executed, direct the computing system to derive a final result of the input message processing from at least one result of the sub-expression processing.

21. (Currently Amended) The computer-readable storage medium ~~one or more computer-readable storage media~~ as recited in claim 19, further comprising computer-executable instructions that, when executed, direct the computing system to perform acts comprising:

determine if a first of the two or more sub-expressions evaluates true;

proceed with processing of subsequent sub-expressions of the two or more sub-expressions if the first sub-expression evaluates to true; and

forego processing of subsequent sub-expressions of the two or more sub-expressions if the first sub-expression evaluates to false.

22. (Currently Amended) The computer-readable storage medium ~~one or more computer-readable storage media~~ as recited in claim 15, wherein each filter sub-engine includes a set of queries against which input messages directed to the respective filter sub-engine are tried, and wherein each set of queries is unique.

23. (Currently Amended) A message processing system[[,]] comprising:

means for receiving a message;

an optimized filter sub-engine that supports only a subset, less than the whole, of a message language, wherein the message conforms to the message language;

a general filter sub-engine that supports all of the message language;

analyzing means for analyzing the message to determine if the optimized filter sub-engine is configured to process the message; and

distribution means for distributing the message:

to the optimized filter sub-engine if the optimized filter sub-engine can process the message; or

to the general filter sub-engine if the optimized filter sub-engine cannot process the message.

24. (Previously Presented) The message processing system as recited in claim 23, wherein:

the optimized filter sub-engine comprises a first set of queries against which the message can be compared;

the general filter sub-engine comprises a second set of queries against which the message can be compared; and

the first set of queries contains fewer queries than the second set of queries.

25. (Previously Presented) The message processing system as recited in claim 23, wherein:

the message language comprises an XML query language;

the general filter sub-engine is configured to support the entire XML query language; and

the optimized filter sub-engine is configured to support a subset of the XML query language, wherein the subset of the XML query language is less than the entire XML query language.

26. (Original) The message processing system as recited in claim 25, wherein the XML query language is XPath.

27. (Previously Presented) The message processing system as recited in claim 23, wherein the optimized filter sub-engine comprises means for increasing message processing performance by combining individual filters for use in a single procedure.

28. (Previously Presented) The message processing system as recited in claim 27, wherein the means for increasing message processing performance further comprises a hash function.

29. (Previously Presented) The message processing system as recited in claim 23, wherein:

the optimized filter sub-engine comprises:

a first optimized filter sub-engine that supports only a first unique subset of the query language; and

a second optimized filter sub-engine that supports only a second unique subset of the query language; and

each of the first and second unique subsets of the query language are less than that entire query language;

the distribution means is further configured to distribute the message to the second optimized filter sub-engine if the first optimized filter sub-engine cannot process the message but the second optimized filter sub-engine can process the message.

30. (Previously Presented) The message processing system as recited in claim 23, further comprising:

means for parsing the message into constituent sub-expressions;

wherein the analyzing means is further configured to process each of the constituent sub-expressions as an individual message and to evaluate sub-expression processing results to derive a result corresponding to the message.

31. (Original) The message processing system as recited in claim 23, wherein the message is a sub-expression of a parent message.

32. (Previously Presented) The message processing system as recited in claim 23, further comprising means for determining whether a filter in the system is associated with the general filter sub-engine or with the optimized filter sub-engine.

33. (Previously Presented) The method as recited in claim 1, wherein: determining comprises generating a hash of the input data in order to determine if an optimized sub-engine is capable of handling the input data.